Freie Universität Berlin

{tip4u://173} Version 4

Zentraleinrichtung für Datenverarbeitung (ZEDAT)
www.zedat.fu-berlin.de

ACME-Client unter Linux

Dieses Merkblatt richtet sich an Personen, die Linux Server an der Freien Universität Berlin betreiben. Es beschreibt die Verwendung einer Software zur initialen Beantragung und Erneuerung von Server-Zertifikaten.

zedat

Zentraleinrichtung für Datenverarbeitung
Fabeckstraße 32 / 14195 Berlin
Hotline (030) 838 77777 / hilfe@zedat.fu-berlin.de

Server-Zertifikate: ACME-Client unter Linux

Voraussetzung für das Erzeugen von Zertifikaten auf einem Server der Freien Universität Berlin ist die Freischaltung der gewünschten Domains sowie das Erstellen eines so genannten ACME-Accounts im Zertifikatsportal der ZEDAT. Beides wird im Merkblatt Tip4U #171[1] beschrieben.

Es ist empfehlenswert, einen dezidierten ACME-Account für jeden Server bzw. Dienst anzulegen, für den ein Zertifikat genutzt werden soll. Die Daten dieses ACME-Accounts werden dann auf dem jeweiligen Server hinterlegt und können dort auch zur automatischen Verlängerung der Zertifikate verwendet werden. Im Folgenden wird beschrieben, wie ein Zertifikat einerseits initial und andererseits im Verlängerungsfall auf einem Linux-Server eingespielt wird.

Unter Linux gibt es eine Vielzahl an ACME-Clients. Wir stellen hier die Verwendung von `acme.sh`, `certbot` und `mod_md` vor. Prinzipiell sollte aber auch jeder andere Client geeignet sein, sofern dieser „External Account Binding“ unterstützt.

acme.sh

`acme.sh` ist ein Shellscript mit sehr wenigen Abhängigkeiten (`openssl` und `curl` oder `wget` werden benötigt). Es ist daher für den Einsatz bei allen Linux-Varianten geeignet, und das offizielle Git-Repository unter https://github.com/acmesh-official/acme.sh kann für die Installation verwendet werden.

Je nach Art der Installation kann es sein, dass das Konfigurationsverzeichnis mit unnötig weitreichenden Schreib- und Lese-Berechtigungen erstellt würde. Es empfiehlt sich daher, das Verzeichnis vor der ersten Verwendung manuell anzulegen:

```
mkdir $HOME/.acme.sh/
chmod 0700 $HOME/.acme.sh/
```

Technisch kann auch ein anderes Verzeichnis verwendet werden, dieses muss dann aber bei weiteren Aufrufen mit `--home` übergeben werden.

Um mit `acme.sh` ein Zertifikat zu beantragen, muss als Erstes der richtige Server angegeben und ein Account registriert werden. Dazu werden die Daten des ACME-Accounts aus dem Zertifikatsportal benötigt:

```
acme.sh --server https://certificate.zedat.fu-berlin.de/acme/ \
  --set-default-ca
acme.sh --register-account --eab-kid <EAB-KID> \
  --eab-hmac-key <EAB-HMAC>
```

Dabei müssen `<EAB-KID>` und `<EAB-HMAC>` durch die entsprechenden Werte des ACME-Accounts aus dem Zertifikatsportal ersetzt werden.

Nachdem der Account registriert wurde, kann das Zertifikat erzeugt werden. `acme.sh` unterstützt mehrere Möglichkeiten, das Zertifikat zu installieren, es wird im Folgenden aber nur die manuelle Methode vorgestellt.

[1] https://zedat.fu-berlin.de/tip4u_171.pdf

Um das Zertifikat zu erhalten, führen Sie den folgenden Befehl aus:

```
acme.sh --issue --dns manual -d <Domain> --days 333 --reloadcmd <Reload>
```

Dabei bitte <Domain> durch den gewünschten Domainnamen ersetzen. Die Option `-d` kann auch mehrfach angegeben werden, falls mehrere Domainnamen in das Zertifikat aufgenommen werden sollen.

Außerdem bitte <Reload> durch einen Befehl zum Neuladen der Webserverkonfiguration ersetzen (z.B. `systemctl reload apache2.server` für Apache2). Dieser Befehl wird nach der erfolgreichen Ausstellung und auch nach den späteren Erneuerungen des Zertifikates ausgeführt.

Das Zertifikat und der zugehörige private Schlüssel befinden sich danach im Verzeichnis: `$HOME/.acme.sh/<Domain>/`

Um das Zertifikat automatisch zu erneuern, muss anschließend ein Cronjob erstellt werden, in dem `acme.sh --cron` aufgerufen wird. Um einen solchen Cronjob für den aktuellen Benutzer anzulegen, kann man zum Beispiel den folgenden Befehl verwenden:

```
acme.sh --install-cronjob
```

certbot

`certbot` ist ein in Python geschriebener ACME-Client und kann unter vielen Linux-Distributionen aus den Paketquellen installiert werden. Bei Debian heißt das benötigte Paket `certbot`. Bei der Installation wird automatisch ein Cronjob oder ein systemd-Timer angelegt, der regelmäßig überprüft, ob Zertifikate erneuert werden müssen.

Um mit `certbot` ein Zertifikat zu beantragen, muss als Erstes ein Account registriert werden. Dazu werden die Daten des ACME-Accounts aus dem Zertifikatsportal benötigt:

```
certbot --server https://certificate.zedat.fu-berlin.de/acme/ \
  register --eab-kid <EAB-KID> --eab-hmac-key <EAB-HMAC> \
  --agree-tos --no-eff-email --register-unsafely-without-email
```

Dabei müssen <EAB-KID> und <EAB-HMAC> durch die entsprechenden Werte des ACME-Accounts aus dem Zertifikatsportal ersetzt werden.

Nachdem der Account registriert wurde, kann das Zertifikat erzeugt werden. `certbot` unterstützt viele Möglichkeiten, das Zertifikat gleich in der Konfiguration des Webservers anzupassen. Diese können bei Bedarf in der offiziellen Dokumentation nachgelesen werden: https://eff-certbot.readthedocs.io/

Im Folgenden wird der Modus beschrieben, der das Zertifikat nur in ein Verzeichnis speichert. Von dort kann das Zertifikat dann manuell im Webserver eingebunden werden. Um das Zertifikat zu erhalten, führen Sie den folgenden Befehl aus:

```
certbot --server https://certificate.zedat.fu-berlin.de/acme/ \
  certonly --standalone -d <Domain>
```

Dabei bitte <Domain> durch den gewünschten Domainnamen ersetzen. Die Option `-d` kann auch mehrfach angegeben werden, falls mehrere Domainnamen in das Zertifikat aufgenommen werden sollen.

Das Zertifikat und der zugehörige private Schlüssel befinden sich danach in folgendem Verzeichnis: `/etc/letsencrypt/live/<Domain>/`

Von dort aus können sie in den Webserver eingebunden werden. Die Dateien sollten von dort aber möglichst nicht kopiert oder verschoben werden, damit bei einer Zertifikatserneuerung das Zertifikat wieder an der gleichen Stelle abgelegt wird und die Serverkonfiguration nicht geändert werden muss.

Damit bei einer Zertifikatserneuerung das neue Zertifikat automatisch aktiviert wird, muss bei den meisten Webservern die Konfiguration neu geladen werden. Dazu kann in der Datei `/etc/letsencrypt/cli.ini` ein entsprechender Befehl angegeben werden, der dann ausgeführt wird. Für Apache2 könnte dieser so genannte `deploy-hook` zum Beispiel folgendermaßen aussehen:

```
deploy-hook = systemctl reload apache2.service
```

mod_md

`mod_md` ist ein Modul für den Apache2-Webserver zur automatischen Verwaltung von Zertifikaten. Das Modul kann unter verschiedenen Linux-Distributionen aus den Paketquellen installiert werden. Unter Debian wird das Modul sogar standardmäßig mit dem `apache2` Paket installiert.

Damit mit `mod_md` Zertifikate erstellt werden können, muss als Erstes das Modul aktiviert werden:

```
a2enmod md
```

In der globalen Serverkonfiguration (außerhalb der VirtualHost-Angaben) müssen allgemeine Einstellungen für den ACME-Client gesetzt werden:

```
MDCertificateAuthority https://certificate.zedat.fu-berlin.de/acme/
MDExternalAccountBinding <EAB-KID> <EAB-HMAC>
MDCAChallenges tls-alpn-01
MDCertificateAgreement accepted
```

Dabei müssen `<EAB-KID>` und `<EAB-HMAC>` durch die entsprechenden Werte des ACME-Accounts aus dem Zertifikatsportal ersetzt werden.

Außerdem müssen in der Apache-Konfiguration die Domains für die Zertifikate spezifiziert werden:

```
MDomain <Domain>
```

Dabei bitte `<Domain>` durch den gewünschten Domainnamen ersetzen. Die Domain muss der Angabe `ServerName` einer VirtualHost-Konfiguration entsprechen. Es können auch weitere, mit Leerzeichen getrennte Namen angegeben werden, um diese in das gleiche Zertifikat aufzunehmen. Mehrmaliges Setzen von `MDomain` führt hingegen dazu, dass mehrere Zertifikate erzeugt werden. Sind in der VirtualHost-Konfiguration weitere Namen mit `ServerAlias` gesetzt, werden diese automatisch in das Zertifikat aufgenommen.

In der eigentlichen VirtualHost-Konfiguration muss dann nur noch SSL aktiviert, aber kein Zertifikat mehr konfiguriert werden:

```
<VirtualHost *:443>
  ServerName <Domain>
  DocumentRoot htdocs/

  SSLEngine on
</VirtualHost>
```

Nach dem Start von Apache2 mit dieser Konfiguration wird der Webserver im Hintergrund das Zertifikat beantragen und im Dateisystem ablegen; beim nächsten Neustart oder Reload des Servers steht das Zertifikat dann zur Verfügung.

Achtung: Solange kein Zertifikat verfügbar ist, werden Anfragen an die entsprechenden Domains mit `503 Service Unavailable` beantwortet.

`mod_md` bietet auch Möglichkeiten, den Status der Zertifikate abzufragen. Diese und weitere Optionen können bei Bedarf in der offiziellen Dokumentation unter https://httpd.apache.org/docs/mod/mod_md.html nachgelesen werden.